

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено  
Завідувач кафедри

О.В.Коваль

(підпис)

(ініціали, прізвище)

“ ” 2019 р.

**ДИПЛОМНА РОБОТА**  
**на здобуття ступеня бакалавра**

з напряму підготовки  
6.050101 “Комп’ютерні науки”

на тему: Інформаційна система біоінформаційного та епідеміологічного аналізу  
вакцинопрофілактики гострих вірусних інфекцій

Виконав: студент 4 курсу, групи ТР-52

Горбатюк Олександр Іванович

(прізвище, ім’я, по батькові)

(підпис)

Керівник ст. викл. Бандурка Олена Іванівна

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент

(підпис)

Київ – 2019

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Горбатюку Олександрю Івановичу

(прізвище, ім’я, по батькові)

1. Тема роботи “Інформаційна система біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій”

керівник роботи \_\_\_\_\_ ст. викл. Бандурка Олена Іванівна

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” \_\_\_\_ ” \_\_\_\_\_ 201\_\_ р.  
№ \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_ 201\_\_ р.

3. Вихідні дані до роботи персональний комп’ютер під керуванням операційної системи Microsoft Windows, мова програмування JavaScript, мова розмітки HTML, мова стилів CSS, фреймворк Electron.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_ проаналізувати існуючі програмні рішення та можливі засоби реалізації взаємодії, обґрунтувати обрані програмні застосунки та шляхи розробки програмних додатків, розробити програмне забезпечення, розробити користувацький інтерфейс, зробити висновки за результатами роботи

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов’язкових креслень)

1. Актуальність дослідження, мета 2. Епідеміологічна модель. 3. Епідеміологічна модель у вигляді різницевого рівнянь. 4. Основні блоки системи. 5. Використані

програмні засоби. 6.Висновки.

6. Публікації: XVII Міжнародна науково-практична конференція аспірантів, магістрантів і студентів “Сучасні проблеми наукового забезпечення енергетики”, м. Київ, КПІ ім. Ігоря Сікорського, 23-26 квітня 2019 року.

Дата видачі завдання ” \_\_\_\_ ” \_\_\_\_\_ 201\_\_ р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі		
2.	Розробка архітектури та загальної структури системи		
3.	Розробка структур окремих підсистем		
4.	Підготовка матеріалів		
5.	Програмна реалізація системи		
6.	Захист програмного продукту		
7.	Оформлення пояснювальної записки		
8.	Передзахист		
9.	Захист		

Студент

\_\_\_\_\_  
(підпис)

Горбатюк О.І.

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Бандурка О.І.

\_\_\_\_\_  
(прізвище та ініціали)

# АНОТАЦІЯ

Мета роботи — реалізувати інформаційну систему для ефективного аналізу вакцинопрофілактики гострих вірусних інфекцій. Програма надає можливість визначати ефективність вакцини проти конкретного типу вірусу в спостереженій групі населення та моделювати поведінку епідеміологічної системи з різними вхідними параметрами. Користувач матиме змогу побачити розвиток інфекції без втручання у вигляді вакцинації, з повною або частковою вакцинацією населення. Таким чином спеціалісту буде надана можливість опрацьовувати дані різних популяцій людей з урахуванням розвитку вірусної інфекції та прогнозувати подальшу поведінку групи населення.

Записка містить 70 сторінок, 21 рисунок, 4 формули, 4 додатки і 17 посилань.

Ключові слова: JAVASCRIPT, ELECTRON, ЕПІДЕМІОЛОГІЧНА МОДЕЛЬ, ВАКЦИНОПРОФІЛАКТИКА, ВІРУС, ШТАМ, ПРОФІЛАКТИКА.

# ABSTRACT

The purpose of the work is to develop an information system for effective analysis of vaccine prevention of acute viral infections. The program gives an opportunity to define the effectiveness of the vaccine against the certain type of virus in the observed group of people and model the behavior of the epidemiological system with different input parameters. The user will be able to see the progress of infection without interfering in the form of vaccination, with full or partial vaccination of the population. In this way the specialist will be given an opportunity to handle the data of different populations of people considering the progress of virus infection and predict further behavior of the population

The note contains 70 pages, 21 figures, 4 formulas, 4 attachments and 17 links.

KEYWORDS: JAVASCRIPT, ELECTRON, EPIDEMIOLOGICAL MODEL, VACCINATION, VIRUS, STRAIN, PREVENTION.

# ЗМІСТ

Перелік умовних позначень, скорочень і термінів.....	6
ВСТУП.....	7
1 АНАЛІЗ ПРОБЛЕМИ БІОІНФОРМАЦІЙНОГО ТА ЕПІДЕМІОЛГІЧНОГО АНАЛІЗУ ВАКЦИНОПРОФІЛАКТИКИ ГОСТРИХ ВІРУСНИХ ІНФЕКЦІЙ .....	9
1.1 Сімейство програмних продуктів MS Office.....	12
1.2 Сімейство програмних продуктів MatLab .....	13
1.3 Система MathCad .....	14
Висновки до розділу 1 .....	14
2 ЗАСОБИ РОЗРОБКИ .....	15
2.1 Середовище розробки Visual Studio Code .....	15
2.2 Мова програмування JavaScript.....	16
2.3 Фреймворк Electron .....	17
Висновки до розділу 2.....	18
3 ОПИС ЕПІДЕМІОЛОГІЧНОЇ МОДЕЛІ ТА ВИКОРИСТАНИХ МЕТОДІВ....	19
3.1 Когортна модель .....	19
3.2 Огляд епідеміологічних моделей .....	20
3.3 Переваги та недоліки існуючих моделей .....	23
3.4 Епідеміологічна модель розроблюваної системи .....	24
3.5 Метод Монте-Карло .....	26
3.6 Марківський процес прийняття рішень.....	27
Висновки до розділу 3.....	28
4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ .....	29
4.1 Системні вимоги .....	29

4.2	Робота з програмним модулем .....	30
	Висновки до розділу 4.....	40
	ВИСНОВКИ .....	41
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	42
	Додаток А .....	44
	Додаток Б.....	46
	Додаток В .....	50
	Додаток Г .....	59

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

HTML — Hyper Text Markup Language.

CSS — Cascading Style Sheets.

SPA — Single Page Application.

DOM — Document Object Model.

BOM — Browser Object Model.

CORBA — Common Object Request Broker Architecture.

API — Application Programming Interface.

OS – Operation System.

МППР – Марківський Процес Прийняття Рішень.

## ВСТУП

Одним з найважливіших завдань вірусології та епідеміології є прогнозування поширення, розвитку та наслідків вірусних інфекцій. Щороку в Україні реєструють близько 6 млн інфекційних захворювань, 98% з яких припадає на грип та ГРВІ. Програма вакцинації конкретною вакциною в певній віковій групі чи групі ризику скорочує число випадків захворювання.

Епідеміологічні моделі дозволяють досліджувати процес розповсюдження інфекційних захворювань, складати прогноз на майбутнє, визначати ефективність профілактичних заходів (наприклад, вакцинація) [1]. Завдяки використанню епідеміологічних моделей стає можливим своєчасне попередження епідемій інфекційних захворювань [2].

Одна із основних причин, яка ускладнює ліквідацію контрольованих інфекційних захворювань, пов'язана з епідеміологічним парадоксом. Оскільки охоплення вакцинацією збільшує кількість імунізованого населення, частка захищених осіб перевищує критичний рівень імунізації, вище якого інфекційне захворювання більше не може зберігатися. Цей ефект називається колективним популяційним імунітетом. При досягненні колективного імунітету, невакциновані особи менш ймовірно стануть інфікованими, оскільки вони опосередковано захищені кількістю вакцинованих осіб. Таким чином, невакциновані особи отримують користь від колективного імунітету без урахування ризиків, пов'язаних з вакцинацією, таких як ускладнення, побічні ефекти та фінансові витрати. Існує менший стимул, для вакцинації, і тоді виникає проблема так званого «вільного воїна». Деякі дослідження показують, що благополуччя суспільства може бути під загрозою, якщо надто багато людей сприймають колективний імунітет як суспільне благо [3]. Внаслідок цього занадто особистий інтерес дестабілізує стан колективного імунітету і хвороба починає пробуджуватись, наступати. Цей парадокс надто



ускладнює повну ерадикацію захворювання за принципом добровільної вакцинації та викликає конфлікт між оптимальним ефектом вакцинації для кожної особи та достатнім рівнем вакцинації, необхідним для захисту всього суспільства за допомогою популяційного імунітету [4].

Крім вакцинації, можуть також існувати інші немедикаментозні втручання, які можуть допомогти зменшити ризики і, що видається особливо важливим, перш ніж ефективні вакцини стануть широко доступні. Тим не менше, більшість існуючих досліджень вважає, що вакцинацію та немедикаментозне втручання треба розглядати окремо.

Слід зазначити, що існує компроміс між двома втручаннями. При збільшенні охоплення вакцинацією, зменшення захворюваності і в результаті зміни в структурі передачі збудника інфекції дозволяють зменшити немедикаментозні втручання. Аналогічним чином, якщо немедикаментозні втручання стають все більш поширеними, охоплення вакцинацією буде зменшуватися приблизно в однаковій мірі, навіть якщо представляється неможливим одночасно підвищити ефективність обох заходів. Крім того, зусилля по скороченню поширення інфекції шляхом розширення немедикаментозних заходів майже повністю нівелюються в результаті зниження рівня охоплення вакцинацією, в той час як зусилля по розширенню охоплення вакцинацією слабо залежать від ефекту не лікувальних втручань.

Таким чином, відсутність вакцинації є значнішою, ніж немедикаментозних заходів в зниженні захворюваності грипу. При розгляді декількох програм втручання необхідно звернути особливу увагу на потенційні перешкоди, зокрема, з огляду на дисасортативність між вакцинацією і немедикаментозними втручаннями в реальних життєвих ситуаціях. Важливо донести до громадськості, що ці заходи зазвичай доповнюють один одного.

# **1 АНАЛІЗ ПРОБЛЕМИ БІОІНФОРМАЦІЙНОГО ТА ЕПІДЕМІОЛОГІЧНОГО АНАЛІЗУ ВАКЦИНОПРОФІЛАКТИКИ ГОСТРИХ ВІРУСНИХ ІНФЕКЦІЙ**

Соціальні мережі та поширення інфекційних захворювань в такій мережі є інструментами дослідження епідеміологічного аналізу та вакцинопрофілактики, оскільки складність, заплутаність що лежить в основі організації соціальної мережі служить ключовим фактором при визначенні рівноваги. Вакцинація, яка обумовлює індивідуальну несприйнятливість до вірусних інфекцій, гальмує її сприйняття. Крім того, існує популяційний імунітет, тобто певна особа, яка не вакцинована, може не бути інфікованою. Взагалі, існує конфлікт між індивідуальними та соціальними вигодами від вакцинації; словом, конфлікт між індивідуальним раціональним вибором – намагатися уникнути вакцинації, або погодитись на неї, досягнувши справедливого оптимального паритету. Вакцинацію в такому випадку можна розглядати як гру в складній соціальній мережі [5].

Вакцинація, зазвичай є добровільною, тому прийняття рішень на індивідуальному рівні може бути результатом компромісу між захистом та усвідомленням ризиків й витрат на вакцинацію [6].

Єдиним прикладом повністю успішної програми вакцинації, яка призвела до певної ерадикації захворювання була вакцинація проти натуральної віспи, тоді як циклічні (сезонні) підйоми захворювання інших інфекційних захворювань, таких як гострі кишкові та респіраторні вірусні захворювання, залишаються серйозною проблемою.

Одна із основних причин, яка ускладнює ліквідацію контрольованих інфекційних захворювань, пов'язана з епідеміологічним парадоксом. Оскільки

охоплення вакцинацією збільшує кількість імунізованого населення, частка захищених осіб перевищує критичний рівень імунізації, вище якого інфекційне захворювання більше не може зберігатися. Цей ефект називається колективним популяційним імунітетом. При досягненні колективного імунітету, невакциновані особи менш ймовірно стануть інфікованими, оскільки вони опосередковано захищені кількістю вакцинованих осіб. Таким чином, невакциновані особи отримують користь від колективного імунітету без урахування ризиків, пов'язаних з вакцинацією, таких як ускладнення, побічні ефекти та фінансові витрати. Існує менший стимул, для вакцинації, і тоді виникає проблема так званого «вільного воїна». Деякі дослідження показують, що благополуччя суспільства може бути під загрозою, якщо надто багато людей сприймають колективний імунітет як суспільне благо [7]. Внаслідок цього занадто особистий інтерес дестабілізує стан колективного імунітету і хвороба починає пробуджуватись, наступати. Цей парадокс надто ускладнює повну ерадикацію захворювання за принципом добровільної вакцинації та викликає конфлікт між оптимальним ефектом вакцинації для кожної особи та достатнім рівнем вікцинації, необхідним для захисту всього суспільства за допомогою популяційного імунітету [8].

Крім того, кількість вакцинованих осіб може бути зменшена через недооцінку ризику інфікування через відсутність знань про захворювання або завищення ризику вакцинації на основі науково необґрунтованої інформації [9]. Взаємовідносини між охопленням вакцинацією, поширеністю захворювання та ставленням людей до вакцинації є складним, і необхідно вивчати, динамічно і кількісно прогнозувати наслідки цих взаємин.

Більшість інфекційних захворювань викликані етіологічним агентом, який може бути безпосередньо переданий від однієї особи до іншої. Їх інфекційна природа має важливі наслідки для ефективності профілактичного втручання, оскільки ризик інфікування залежить не тільки від індивідуальних особливостей сприйнятливої особи, але також і від поширеності інфекції серед інших осіб в популяції, з якими сприйнятлива особа може вступати в контакт. Якщо частка населення, яка є сприйнятною до інфекції, є досить низькою через велику

кількість інфікованих осіб, природний імунітет до інфекції або успішну вакцинацію, то кожна інфікована особа буде причиною менше, ніж одного іншого випадку інфікування до її одужання. Отже, поширеність інфекції буде зменшуватися з плином часу. Колективний або популяційний імунітет веде до зниження швидкості поширення інфекції в популяції, велика частина якої має імунітет, а отже, зниження ймовірності того, що інфікована особа буде мати контакт із сприйнятливими особами. Частка населення, яка залишається сприйнятною, і при якій захворювання не поширюватиметься (бо кожна інфікована особа передає збудник інфекції в середньому менше ніж одній сприйнятливій особі), називається порогом колективного імунітету. Таке явище є важливим при розробці моделей вакцинації, оскільки мета вакцинації полягає в захисті досить великої частини населення, щоб запобігти поширенню захворювання. Це має важливі наслідки для моделювання епідеміологічного процесу інфекційного захворювання. Для інфекційних захворювань, як правило, більш доцільно використовувати динамічні епідеміологічні моделі. Вони представляють усе населення, яке, як правило, є відкритим, хоча така модель протягом дуже короткого часового проміжку може розглядати закрите населення. Крім того, втручання відносно однієї особи може впливати на інших осіб, оскільки ймовірність сприйнятливої особи інфікуватися залежить не тільки від кількості осіб в сприйнятливому стані, але і від кількості інфікованих осіб. Сценарій епідемічного процесу РВІ як гострої кишкової інфекції з точки зору математичної епідеміології найкраще описується моделлю, запропонованою У. Кермаком і А. Маккендріком у 1927 році. Відповідно до цієї моделі осіб в популяції можна розділити на «сприйнятливих» (S - “susceptible”), які раніше не піддавалися впливу збудника), «інфікованих» (I – “infectious”), які є носіями інфекційних агентів з проявом клінічних симптомів або без них), та «тих, що одужали» (R – “recovered”), з набутих специфічним імунітетом і за умови елімінації збудника. Оскільки РВІ може бути перенесена кілька разів особливо впродовж перших років життя дитини, що обумовлено різноманіттям генотипів циркулюючих ротавірусів та відсутністю стійкого перехрестного імунітету, епідеміологічна модель може бути спрощена без втрати якості (рисунк 3.1.1),

зважаючи на те, що одужання після перенесення інфекції та елімінації збудника супроводжується “миттєвим” поверненням до групи сприйнятливих осіб та можливістю повторного випадку захворювання, викликаного ротавірусами іншого серо- або генотипу.

Тому на сьогоднішній день не існує однозначного рішення проблеми вакцинопрофілактики гострих вірусних інфекцій. Кількість факторів які впливають на розповсюдження хвороби, тривалість хвороби, ймовірність того що людина погодиться або не погодиться на вакцинацію неймовірно велика. І тому побудова системи за допомогою якої можна було б вирішити цю проблему є складним завданням, яке потребує дуже уважного вивчення предметної області.

Для забезпечення аналітики та звітування в медичній сфері часто використовують пакет програм MS Office, MatLab та MathCad. Його достатньо для базових підрахунків та їх представлення у вигляді звітів з графічним відображенням.

Програмних продуктів, здатних змодельовати популяцію з урахуванням хвороби та втручання в її розвиток, в наш час у вільному доступі взагалі немає. А це лишається основною задачею перед інформаційною системою, що була описана вище.

Моделювання популяцій та прогнозування їх подальшого розвитку в запропонованому програмному продукті спираються на принципи індивідуально-орієнтовного моделювання.

## **1.1 Сімейство програмних продуктів MS Office**

Продукт Microsoft Office був розроблений корпорацією Microsoft та підтримується операційними системами Microsoft Windows та Mac OS. До складу пакету Microsoft Office входять Microsoft Office Word, Microsoft Office Excel, Microsoft Office Power Point та інші.

Основним програмним застосунком який міг би задовольнити наші потреби в межах поставленої проблеми є Microsoft Office Excel. Дана програма являє собою табличний процесор, що підтримує весь необхідний функціонал для створення електронних таблиць різної складності. За допомогою Excel ми можемо розраховувати деякі коефіцієнти системи, використовуючи вбудовані математичні формули та функції. Проте усе сімейство програмних застосунків Microsoft Office має на меті вирішення більш загальних та рутинних проблем і не може повністю задовольнити таку специфічну предметну область як аналіз вакцинопрофілактики гострих вірусних інфекцій.

До того ж ціна на базовий пакет програмних застосунків MS Office в середньому перевищує відмітку в 100 доларів США, що також є одним з важливих факторів. Крім того, людина яка досі не працювала з цими програмами має обов'язково пройти деякий період навчання для набуття специфічних навичок у користуванні. Тому пакет MS Office ми не можемо розглядати як потенціальне вирішення проблеми описаної вище.

## **1.2 Сімейство програмних продуктів MatLab**

Застосунок MatLab – це пакет прикладних програм для числового аналізу, що має більше одного мільйона користувачів по всьому світу. Був розроблений та впроваджений компанією MathWorks, яка була заснована у 1984 році.

Ця система є дуже зручним засобом для роботи з матрицями, функціями, розрахунку різноманітних алгоритмів та графічного відображення результату. Важливою особливістю є можливість створення робочих оболонок з програмами в інших мовах програмування. Для забезпечення всіх цих функцій MatLab має власну мову програмування, що використовується в даному пакеті.

Проте висока складність у використанні для пересічного користувача комп'ютера, що має невеликий досвід роботи з комп'ютерними змушують нас замислитися над релевантністю використання даного програмного рішення в рамках поставленого завдання.

### 1.3 Система MathCad

Програма MathCad являє собою систему комп'ютерної алгебри за допомогою якою легко та зручно розробляти інтерактивні документи з обчисленнями і візуальним супроводженням. Особливістю є те, що застосунок може використовуватися для колективної роботи. Ця програма була винайдена та спроектована Алленом Раздовим з Массачусетського технологічного інституту (MIT). Постійно оновлюється та вдосконалюється розробниками.

Перевагою MathCad є те, що програма легка у використанні та має простий та інтуїтивний для використання інтерфейс. Так само, як і в MatLab використовується система комп'ютерної алгебри Maple. Програма містить безліч вбудованих функцій та операторів для вирішення різнопланових задач. Має застосунки для роботи з векторами та матрицями, статистичні функції що є корисним для нашої задачі. Також MathCad дозволяє документувати всі розрахунки та обчислення.

Проте, висока ціна розміром від семиста доларів США, велике навантаження даної програми на комп'ютер, можливі складності у використанні та невідповідність цілей MathCad нашій предметній області не дозволяють розробити систему біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій на базі даного продукту.

### 1.4 Висновки до розділу 1

У першому розділі була розглянута проблема біоінформаційного аналізу вакцинопрофілактики гострих вірусних інфекцій та основні програмні застосунки, що могли б бути використані для вирішення даної проблеми.

Було визначено що розроблювана система є дуже розгалуженою та має велику кількість факторів які на неї впливають та потребує детального виичення предметної області. Програми які були розглянуті вище не можуть задовільнит наші вимоги з різних причин, тому вирішення даної проблеми є актуальним завданням.

## 2 ЗАСОБИ РОЗРОБКИ

Важливим чинником, під час розробки програмного продукту, є вибір засобів програмної реалізації та технологій, що будуть використовуватися. Середовищем для розробки інформаційної системи було обрано Visual Studio Code.

Для розробки алгоритмів використовувалась мова програмування JavaScript, а сам додаток був виконаний на базі фреймворку Electron. Інтерфейс користувача виконаний за допомогою технологій HTML та CSS.

Для зчитування вхідної інформації в систему було використано модуль node-xlsx, який дозволяє відкривати Excel файли.

### 2.1 Середовище розробки Visual Studio Code

Середовище розробки Visual Studio Code – засіб для створення, редагування та налагодження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і macOS. За допомогою Visual Studio Code можливо легко переглянути структуру виклику та пов'язані функції, повернені значенні та стан тестування. Також можна провести рефакторинг коду, знайти помилки та отримати варіанти їх усунення[10].

Був випущений 2015 року і з того часу дуже швидко розвивається. VS Code підтримує багато мов програмування серед яких: Python, JavaScript, C#, Go та інші. Також дозволяє завантажувати величезну кількість розширень для покращення процесу розробки та власноруч налаштовувати різноманітні конфігурації. В самому редакторі вбудований термінал, що пришвидшує процес розробки.

Загалом цей редактор коду є дуже зручним та водночас швидким у використанні. Кожен користувач може кастомізувати його під свій стиль процесу розробки.



## 2.2 Мова програмування JavaScript

Мова програмування JavaScript є сучасним засобом розробки та невід'ємною частиною веб сторінок в браузерях. Мультипарадигменна мова програмування яка підтримує об'єктно-орієнтовний, імперативний та функціональний стилі. Являється реалізацією мови ECMAScript. Основні архітектурні риси: динамічна типізація, слабка типізація, атоматичне управління пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов програмування, при розробці була ціль зробити мову схожою на Java, але при цьому легкою для використання та вивчення. Мовою JavaScript не володіє якась компанія або організація, що відрізняє її від ряду мов програмування, що використовуються у веб розробці.

Мова JavaScript є об'єктно-орієнтованою, але використовуване в мові прототипування зумовлює розбіжності в роботі з об'єктами в порівнянні з традиційними клас-орієнтовними мовами. Крім того, JavaScript має ряд властивостей, якими користуються функціональні мови програмування – функції, як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання – що надає мові додаткову гнучкість.

Структурно мову JavaScript можна представити у вигляді об'єднання трьох чітко розрізнених між собою одне від одного частин: ядро (ECMAScript), об'єктна модель браузера (Browser Object Model або BOM), об'єктна модель документа (Document Object Model або DOM). Якщо розглядати JavaScript в відрізнених від браузера умовах, то об'єктна модель браузера та об'єктна модель документа можуть не підтримуватися.

Об'єктна модель браузера – браузер-специфічна частина мови, яка є проміжною частиною між ядром та об'єктною моделлю документа. Основне призначення об'єктної моделі браузера – управління вікнами браузера та забезпечення їх взаємодії. Кожне з вікон браузера представляється об'єктом window, центральним об'єктом DOM. Об'єктна модель браузера на даний момент не

стандартизована, однак специфікація знаходиться в розробці WHATWG та W3C. В рамках об'єктної моделі браузерами зазвичай підтримуються наступні можливості: управління файлами, підтримка затримки в виконанні коду та зациклювання з затримкою, системні діалоги, управління адресом відкритої сторінки, управління інформацією про браузер, управління інформацією про параметри монітору, підтримка роботи з HTTP cookie.

Об'єктна модель документа (DOM) – інтерфейс програмування додатків для HTML та XML-документів. Відповідно до DOM, документ (наприклад веб-сторінка) може бути представлений у вигляді дерева об'єктів, що мають ряд властивостей, які дозволяють проводити з ними різні маніпуляції: генерація та додавання вузлів, отримання вузлів, зміна вузлів, зміна зв'язків між вузлами, видалення вузлів.

Мова програмування JavaScript на сьогодні є найпопулярнішою мовою програмування яка активно розвивається та оновлюється. На її базі було створено безліч гнучких фреймворків та бібліотек, найвідоміші з яких: AngularJS, Vue.js, React.js. З їх допомогою розробляються швидкодіючі SPA. Також JavaScript може використовуватися і при розробці серверної частини додатку за допомогою технології Node.js, що зараз дуже активно використовується. Для настільних додатків було створено NW.js та Electron, які також набули доволі великого рівня популярності.

Зазвичай поряд з технологією JavaScript використовуються HTML (HyperText Markup Language) та CSS (Cascading Style Sheets). Так як JavaScript не має власних методів вводу та виводу, вказані вище технології допомагають взаємодіяти з користувачем та відображати інформацію.

## **2.3 Фреймворк Electron**

Фреймворк Electron розроблений GitHub для того щоб вирішити проблему створення нативних графічних додатків для настільних операційних систем з допомогою веб-технологій[11]. Electron включає в себе Node.js, для роботи з back-end, та бібліотеку рендерингу з Chromium.

Ця система дозволяє створювати кросплатформенні додатки(для Windows, Linux, macOS) за допомогою таких технологій як: HTML, CSS та JavaScript.

В Electron процес, який запускає скрипт “main” з “package.json” називається основним процесом. Скрипт, запущений в основному процесі може відображати GUI шляхом створення веб-сторінок. В додатках Electron завжди є лише один головний процес, але не більше.

Так як Electron використовує Chromium для відображення веб-сторінок, Chromium мульти-процесорна архітектура також використовується. Кожна веб-сторінка електрон виконується у власному процесі, котрий називають процес візуалізації. Користувачі Electron мають можливість використовувати API Node.js на веб-сторінках, дозволяючи взаємодіяти на нижньому рівні операційної системи.

## **Висновки до розділу 2**

У даному розділі були описані технології, що використовувалися при розробці програмного продукту. Розглянуті основні їх особливості та місце у сучасному світі технологій розробки програмного забезпечення. Розроблений додаток на мові JavaScript кросплатформений тому може використовуватися на Windows, MacOS та Linux.

### 3 ОПИС ЕПІДЕМІОЛОГІЧНОЇ МОДЕЛІ ТА ВИКОРИСТАНИХ МЕТОДІВ

У процесі вивчення предметної області та розробки програмного продукту були вивчені та проаналізовані деякі специфічні алгоритми та методи, була визначена оптимальна модель для побудови епідеміологічної системи. Метою цього розділу є описати використані алгоритми і методи та їх призначення в програмі.

#### 3.1 Когортна модель

Когортна модель представляє собою формалізоване уявлення про перехід осіб одного віку зі сприйнятливого стану в інфікований та надає можливість визначити скільки разів може захворіти певна особа на РВІ за певний період життя, наприклад від народження і до п'яти років. Це може бути важливим при вирішенні задачі оптимізації витрат на лікування і вакцинопрофілактику РВІ. На рисунку 3.1 зображено схему когортної моделі.

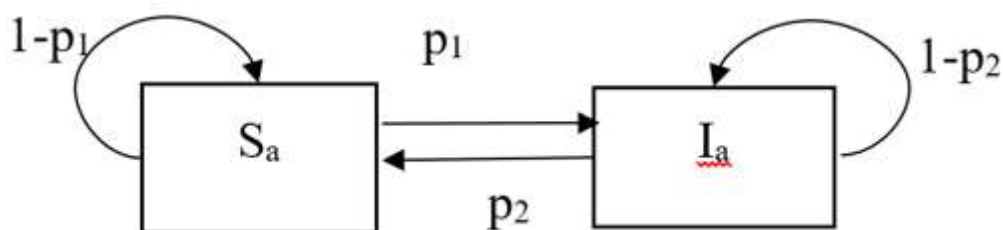


Рисунок 3.1 – Когортна модель РВІ

Де  $p_1$  – ймовірність інфікування протягом часового інтервалу;

$p_2$  – ймовірність одужання протягом часового інтервалу;

$a$  – вікова група.

Представлена модель розглядає досліджувану групу осіб в частках:

$$s = \frac{S}{N}, i = \frac{I}{N}, n = s + i = 1 \quad (3.1)$$

Математично модель може бути записано за допомогою системи рівнянь [F. Bass, A new product growth for model consumer durables, Management Sciences 15 (1969) 215–227.]:

$$\begin{cases} s_{a+1} = (1 - p_{1a}) \cdot s_a + p_{2a} \cdot i_a \\ i_{a+1} = p_{1a} \cdot s_a + (1 - p_{2a}) \cdot i_a \end{cases} \quad (3.2)$$

### 3.2 Огляд епідеміологічних моделей

Епідеміологічні моделі дозволяють досліджувати процес розповсюдження інфекційних захворювань, складати прогноз на майбутнє, визначати ефективність профілактичних заходів (наприклад, вакцинація). Завдяки використанню епідеміологічних моделей стає можливим своєчасне попередження епідемій інфекційних захворювань.

Основний принцип побудови епідеміологічної моделі для конкретного інфекційного захворювання: все населення ділиться на декілька категорій, модель є динамічною системою, яка змінюється в часі, тому при проходженні певного часового періоду частина населення переміщується з однієї категорії в іншу. В епідеміологічних моделях зустрічаються такі категорії:  $S$  – сприятливі особи: чутливі до інфікування.  $I$  — інфіковані особи: заражені інфекційною хворобою, яка представляє інтерес, здатні заражати інших.  $R$  — імунні особи які отримали імунітет або з інших причин недоступні для інфікування.  $V$  — вакциновані особи, отримали імунітет, недоступні до інфікування.

До основних категорій належать: сприйнятливі особи  $S$  та інфіковані особи  $I$ . Модель може мати додаткові категорії: вилікувані (або вилучені) особи  $R$ ,

вакциновані особи  $V$ . В залежності від конкретного інфекційного захворювання, його епідеміологічна модель може мати певні особливості, тому існують різні види моделей.

Перехід осіб із однієї категорії населення в іншу характеризується за допомогою параметрів, наприклад: сила інфекції, швидкість одужання, народжуваність; смертність, міграція тощо. Сила інфекції  $\lambda$  — ймовірність того, що певна особа буде інфікована в певний момент часу. Швидкість одужання  $\gamma$  — ймовірність того, що певна особа буде вилікувана у певний момент часу.

Основні види епідеміологічних моделей —  $SI$  (епідемія, в якій інфіковані особи  $I$  ніколи не виліковуються),  $SIS$  (епідемія, в якій інфіковані особи  $I$  виліковуються, але знову стають сприйнятливими  $S$  (наприклад, гонорея)),  $SIR$  (епідемія, в якій інфіковані особи  $I$  виліковуються та отримують імунітет  $R$  (наприклад, кір, грип)).

Використання  $SIS$ -моделі передбачає, що інфіковані особи одужають по проходженні певного часу і знову стають сприйнятливими до захворювання. На рисунку 3.2 зображено схему  $SIS$ -моделі.

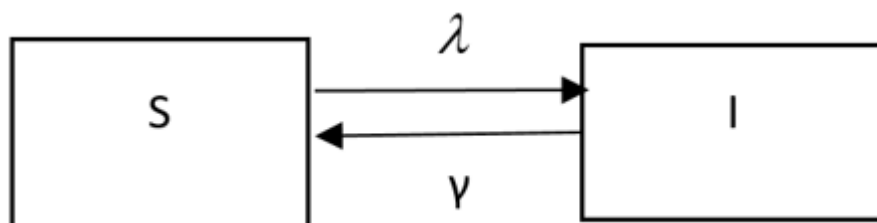


Рисунок 3.2 —  $SIS$ -модель

Де  $N$  — все населення;

$S$  — сприйнятливі особи;

$I$  — інфіковані особи;

$\lambda$  — сила інфекції;

$\gamma$  — швидкість одужання

Параметри моделі, як і вся система, можуть бути динамічними. Значення параметрів на кожному кроці залежать від значень параметрів моделі на минулому

кроці. Математично модель може бути описано за допомогою системи різницьових рівнянь (формула 3.3):

$$\begin{cases} S_{t+1} = S_t - \lambda_t \cdot S_t + \gamma_t \cdot I_t \\ I_{t+1} = I_t + \lambda_t \cdot S_t - \gamma_t \cdot I_t \\ N_t = S_t + I_t \end{cases} \quad (3.3)$$

Де  $t$  – період часу.

Значення невідомих параметрів  $\lambda_t$  і  $\gamma_t$  визначаються на основі системи рівнянь (3.3).

Отримана математична модель є ітеративною, що дозволяє використати її для прогнозування захворювань на визначений період. Щоб скласти прогноз захворюваності необхідно скласти прогноз значень параметрів  $\lambda_t$  і  $\gamma_t$ . Параметри  $\lambda_t$  і  $\gamma_t$  можуть бути або постійними величинами, або мати періодичний характер, що дозволяє спрогнозувати їхні майбутні показники.

Для дослідження впливу стратегій вакцинації та пошуку оптимального охоплення використовується досліджена вище епідеміологічна SIS-модель з вакцинацією, тобто SISV-модель (рисунок 3.3).

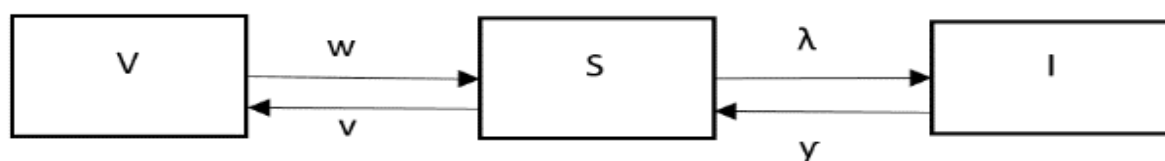


Рисунок 3.3 – SISV-модель

На кожному кроці моделювання деяка кількість людей переходить зі сприятливого до вакцинованого стану. Кількість людей що перейде до вакцинованого стану залежить від ефективності вакцини проти даного вірусу. Також на кожному кроці є деяка кількість людей що повертається до сприятливого стану з вакцинованого стану. Кількість людей, що повертаються до сприятливого стану залежить від параметру  $\omega$ .

Математично модель може бути описано за допомогою системи рівнянь (формула 3.2):

$$\begin{cases} S_{t+1} = S_t - v_t \cdot S_t + w_t \cdot V_t - \lambda_t \cdot S_t + \gamma_t \cdot I_t \\ I_{t+1} = I_t + \lambda_t \cdot S_t - \gamma_t \cdot I_t \\ V_{t+1} = V_t + v_t \cdot S_t - w_t \cdot V_t \end{cases} \quad (3.4)$$

Де  $V_t$  – вакциновані особи;

$\omega_t$  – швидкість спадання імунітету при вакцинації;

$v_t$  – охоплення осіб вакцинацією.

Таким чином, за допомогою *SISV*-моделі можна визначити оптимальну схему вакцинації населення для запобігання розповсюдження інфекційних захворювань. За допомогою моделі можливо знайти оптимальні індивідуальні витрати на вакцинацію (рівень витрат на вакцинацію, за якого окрема особа згодна оплатити витрати на вакцинацію), оптимальних суспільних витрат на вакцинацію, якщо відома вартість лікування однієї особи, вартість вакцинації однієї особи.

### 3.3 Переваги та недоліки існуючих моделей

Існує два найпоширеніших підходу до моделювання: агентного підходу та підходу, що базується на системі диференціальних рівнянь.

Основна модель перебігу захворювання протікає через стани чутливе-інфіковане-відновлене, яка описується диф-рівняннями. Проста модель буде припускати постійний рівень інфікування та швидкість відновлення, часто встановлюючи чисельність населення як контрольовану постійну.

Модель на основі агента працює дуже схожим чином, що дозволяє здійснювати ті ж переходи, але на відміну від диференціальної моделі агентна модель робить рішення для кожної окремої людини окремо, чи вона рухається, чи залишається в певному стані.



Одною з переваг агентної моделі – це стохастичний рух. Перевага цього полягає в тому, щоб забезпечити більш реалістичну поведінку в цілому та при виникненні екстремальних подій. При налаштуванні індивідуалізованих параметрів для населення в агентній моделі, динаміку населення потрібно буде розглянути під полем імовірності. Природа моделі на основі агента вимагатиме переходу від станів до функціонування всього населення, отже часто існує великий обчислювальний час. Це, як правило, не є проблемою моделей диференціальних рівнянь, оскільки використовується середня динаміка населення. Але ж при прийнятті рішень з точки зору оптимальної методології вибору, важливо мати на увазі обчислювальний час, який вимагає агентна модель, у порівнянні з диференціальною моделлю, а саме відносно спрощені агентні моделі вимагають до 600 годин.

Диференціальні моделі малоефективні з дуже малими інфекціями та менш ніж одним новим індивідумом, зараженим в одну часову одиницю. Тому такі моделі вимагають розподіл припущень, які мають великий вплив на початкові фази, і можуть мати великі відмінності в темпах прогресування захворювання.

### 3.4 Епідеміологічна модель розроблюваної системи

Ця *SISV*-модель є розширеною та вдосконаленою. Усе населення в даній моделі може поділятися на дві підгрупи. Одна з підгруп вакцинується, інша – ні. Кожу з підгруп складає деяка частина сприятливих та інфікованих людей ( $S1$ ,  $I1$  та  $S2$ ,  $I2$ ). Величину цих підгруп по відношенню одна до одної ми корелюємо за допомогою коефіцієнтів  $m$  та  $n$ , а наявність чи відсутність самого розділення на підгрупи ми визначаємо перемикачем  $c2$ . За допомогою перемикача  $c1$  ми визначаємо чи є система закритою чи відкритою, тобто чи матимемо ми певний притік та відтік населення на кожному кроці моделювання. Перемикач  $c3$  визначає як саме буде проходити вакцинація населення: на вході в систему, чи в середині системи. Також наявний притік населення на кожному кроці та вихід деякої частини людей з системи.

У своїй роботі я буду використовувати *SISV*-модель (рисунок 3.4).

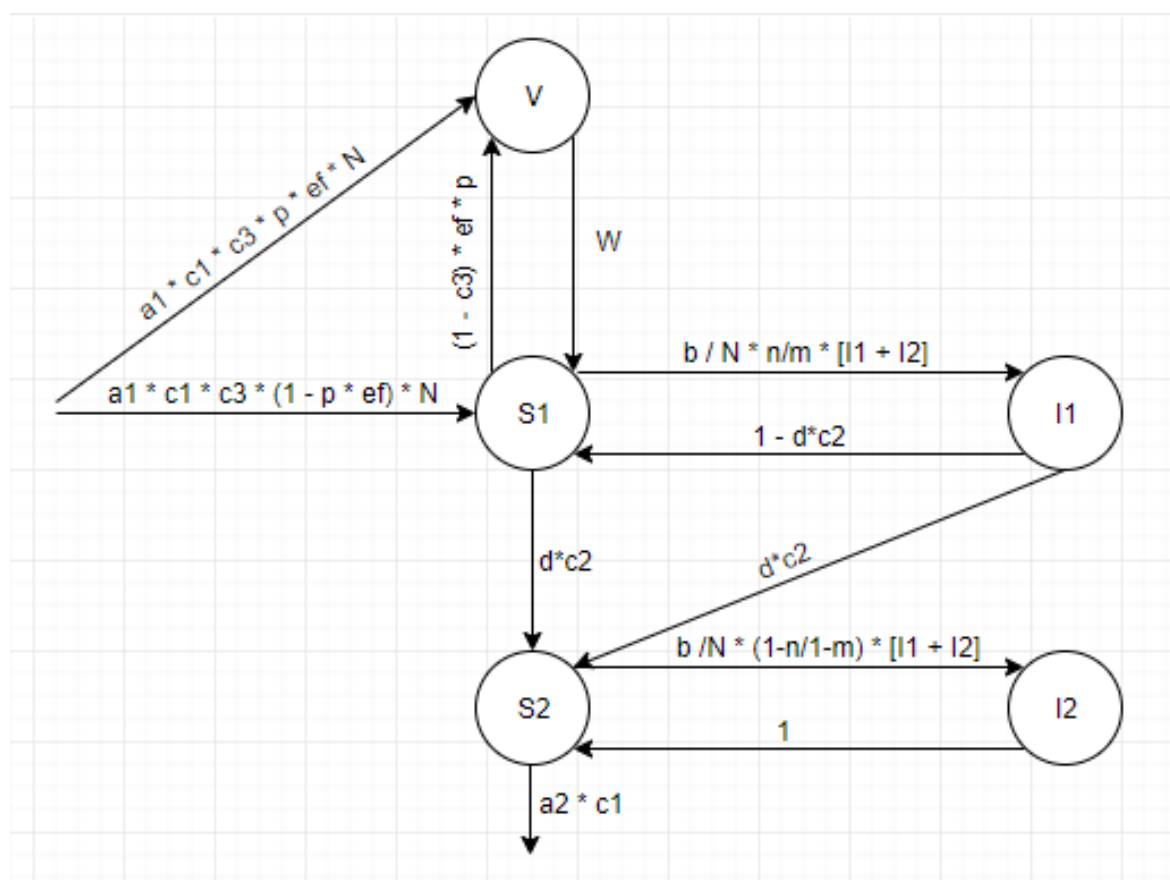


Рисунок 3.4 – Епідеміологічна модель РБІ (*SISV*-модель) з урахуванням повторних випадків інфекції та демографічної динаміки з вакцинопрофілактикою

Параметри епідеміологічної моделі:

- $S$  – сприятливе до хвороби населення;
- $I$  – вражене хворобою населення;
- $N$  – загальний обсяг населення;
- $V$  – вакциноване населення;
- $a_1$  – притік населення;
- $a_2$  – відтік населення;
- $ef$  – ефєктивність вакцини;
- $d$  – коефіцієнт переходу з однієї підгурпи у іншу;
- $p$  – обсяг вакцинації;
- $n, m$  – визначає розмір підгрупи;

- $b$  – параметр передачі вірусного збудника;
- $c1$  – визначає відкритість або закритість системи;
- $c2$  – визначає динаміку всередині системи;
- $W$  – вірогідність повернення людини до сприятливого стану;
- $c3$  – визначає тип вакцинації (на вході в систему або всередині системи).

Таким чином, через наявність великої кількості чинників на модель вона є дуже гнучкою та може легко змінювати свою поведінку.

### 3.5 Метод Монте-Карло

Методи Монте-Карло – це широкий клас обчислювальних алгоритмів, які засновані на повторюванні генерації випадкової вибірки для отримання чисельних результатів. Їх основною ідеєю є використання випадковості для вирішення проблем, які можуть бути детерміністичними в принципі. Вони часто використовуються в фізико-математичних задачах і є найбільш корисними, коли важко або неможливо використовувати інші підходи. Методи Монте-Карло переважно використовуються в трьох проблемних класах: оптимізація, чисельне інтегрування та генерація рисунків з розподілу ймовірності.

В принципі, методи Монте-Карло можуть бути використані для вирішення будь-якої проблеми, що має імовірнісну інтерпретацію. За законом великих чисел, інтеграли, що описуються очікуваним значенням деякої випадкової величини, можуть бути апроксимовані, беручи емпіричне значення незалежних вибірок змінної. Коли розподіл імовірності змінної параметризується, математики часто використовують методи Марківських ланцюгів Монте-Карло. Головною ідеєю є розробка розумної моделі Марківського ланцюга з заданим стаціонарним розподілом імовірності. Тобто вибірки, що генеруються даним методом, будуть згенеровані з бажаного (цільового) розподілу.

### 3.6 Марківський процес прийняття рішень

Марківський процес прийняття рішення (МППР) (рисунок 3.5) – це метод для опису та аналізу послідовних рішень в умовах невизначеності. Він структурно схожий на марківський ланцюг за винятком того, що матриця переходів залежить від дій або стратегій особи, що приймає рішення, в кожний момент часу. Моделі МППР також відомі як послідовні стохастичні оптимізації, дискретно-часові процеси стохастичного керування, стохастичного динамічного програмування тощо. Метою особи, що приймає рішення, є визначення стратегій, при якій серія рішень є оптимальною. Цей підхід знайшов місце в дослідженні операцій, починаючи з 60-их років 20-го сторіччя. МППР був застосований в широкому діапазоні областей, таких як управління запасами, фінансами, мереж зв'язку, водними ресурсами, теорії ігор, машинного навчання і нещодавно в охороні здоров'я.

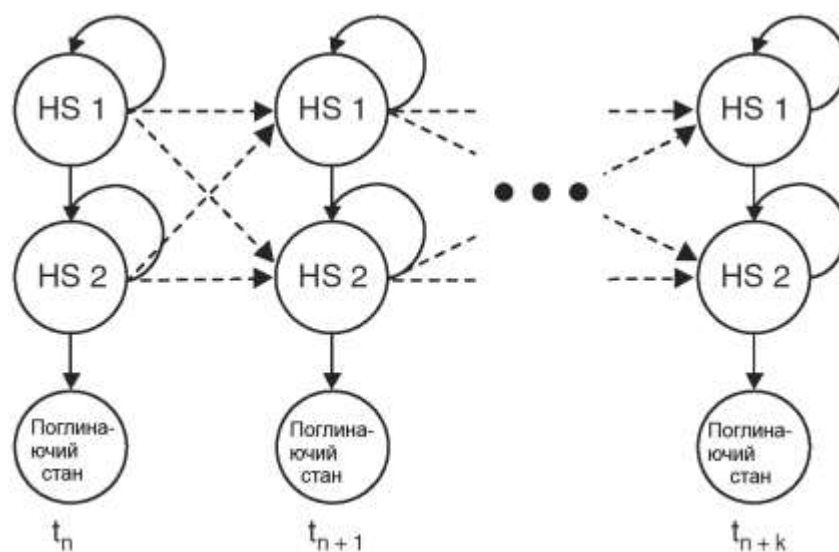


Рисунок 3.5 – Марківський процес прийняття рішень

У МППР дії особи, що приймає рішення, задаються стратегією, яка визначає реакцію на всі можливі стани системи в кінці кожного часового інтервалу за весь часовий горизонт. Вибір дій залежить від виграшу, тобто чистого прибутку попередньої ітерації.

Кожна дія має структуру винагороди, таку як очікуване значення результату рішення в стохастичній марківській моделі. У часі  $t_{n+1}$  стан системи змінюється. Ця зміна є імовірнісною і марківською, тобто, новий стан системи заснований виключно на стані системи в часі  $t_n$ . Метою МППР є визначення стратегії, яка максимізує винагороду або оптимум серед декількох параметрів, таких як тривалість життя та витрати. Розробник МППР може скористатися тим, що правила прийняття рішення можуть бути динамічно запрограмованими. Крім того, правила можуть бути змінені на основі зворотних зв'язків, на основі отриманих виграшів в кінці кожного циклу для пошуку оптимальних рішень.

Одним з ключових припущень, зроблених в МППР є те, що особа, яка приймає рішення, повинна обрати стратегію в рамках всього часового горизонту моделювання. Більшість справжніх осіб, що приймають рішення не схильні до такої поведінки. Динамічне програмування може допомогти визначити оптимальну стратегію прийняття рішення і більш адекватну модель поведінки реальних осіб, які приймають рішення (тобто адаптації існуючої стратегії, щойно нова інформація стає доступною). Тим не менш, ці ж адаптивні методи, за допомогою наприклад, нейронних мереж, можуть приховувати оптимальний результат для даного набору правил, які можуть зруйнувати основну мету використання моделі. І, нарешті, в даний час обмеженою є кількість доступного програмного забезпечення, готового до використання менш досвідченими користувачами.

### 3.7 Висновки до розділу 3

У даному розділі розглянуто існуючі епідеміологічні моделі, обрана та вдосконалена та модель яка найкраще підходить для нашої задачі. Були описані ключові методи що використовуються при реалізації алгоритму програми.

## **4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ**

В цьому розділі описано вимоги до технічного забезпечення та методику роботи користувача з програмою.

### **4.1 Системні вимоги**

Для забезпечення стабільної роботи програми для аналізу вакцинопрофілактики гострих вірусних інфекцій комп'ютер на якому буде використовуватися програмне забезпечення має відповідати наступним вимогам:

1. Операційна система від Windows 7 та вище, MacOS або Linux.
2. Вільна пам'ять на жорсткому диску мінімум 160 Мб.
3. Процесор Intel Pentium та вище з тактовою частотою 1.4 ГГц.
4. Відеокарта AMD Radeon Graphics та краще.
5. Операційна пам'ять на комп'ютері від 1 Гб.

При роботі з програмою користувачу знадобляться пристрої вводу та виводу. Також для виконання розрахунків в додатку користувач повинен мати на комп'ютері програму Microsoft Excel або хочаб заздалегідь підготовлені файли з розширенням .xlsx, які містять необхідну епідеміологічну інформацію.

Також для роботи програми повинен бути підключений інтернет для виведення графіків, адже в роботі використовується API Google Charts.

Ніякого додаткового програмного забезпечення для роботи програми не потребується.

## 4.2 Робота з програмним модулем

Після запуску виконуваного файлу системи користувач побачить головне вікно додатку “Вакцинопрофілактика” (рисунок 5.1). Програма містить три головні блоки в яких виконуються розрахунки: “Аналіз динаміки збудника”, “Розрахунок параметру передачі вірусного збудника”, “Побудова епідеміологічної моделі”.



Рисунок 5.1 – Головне вікно додатку “Вакцинопрофілактика”

Спершу користувач має звернутися до першого блоку під назвою “Аналіз динаміки збудника”, який знаходиться зліва (рисунок 5.2). Для роботи з цим блоком програми користувач повинен мати підготовлений Excel файл інформація з якого знадобиться для розрахунків в програмі, тому він має бути Цей файл має містити інформацію щодо вірусної інфекції, моделювання поведінки якої буде виконувати програма, а саме – розподіл в долях груп збудників хвороби з гомологією за вказаним генотипом.

Файл повинен мати наступну структуру: обов’язково повинна бути колонка під назвою “Vik”, яка відображає віковий проміжок часу на якому спостерігалась

поведінка вірусу, кожна за наступних відноситься до окремого генотипу та відображає його частку по відношенню до всіх генотипів (штамів) даного вірусу.

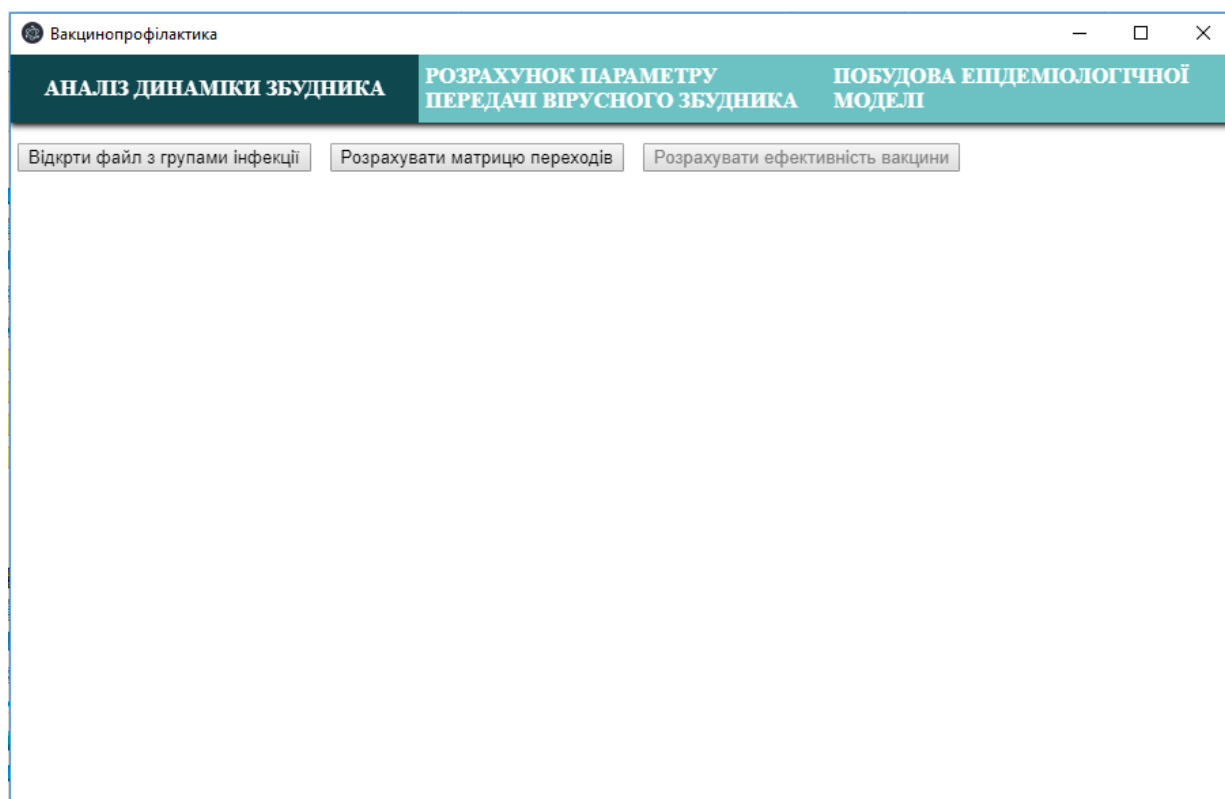


Рисунок 5.2 – Перший блок програми під назвою “Аналіз динаміки вірусного збудника”

Колонки генотипів бажано мають називатися так само як і генотип до якого вони відносяться. Кількість колонок з генотипами може бути різною в залежності від вірусу. Всі колонки мають містити числа. Приклад Excel файлу для використання у програмі (рисунок 5.3).

Коли користувач впевнився у тому що структура його Excel файлу відповідає поставленим вимогам він має натиснути кнопку “Відкрити файл з групами інфекції” та вибрати на своєму комп’ютері документ відповідного змісту. Після цього натиснути кнопку “Відкрити”. Всі колонки мають йти одна за одною без пропусків, кожна комірка також має бути заповнена, якщо частка з певним генотипом вірусу на даному кроці відсутня, то має бути записаний нуль до комірки. Після вдалого завантаження Excel документу користувач побачить його вміст у себе на екрані



(рисунок 5.4), таким чином зможе впевнитися що він відкрив потрібний йому документ.

	A	B	C	D	E	F
1	Vik	G1P8	G1Pother	Gotherp8	OtherOther	
2	0	0.01	0.684	0	0.306	
3	1	0	0.58	0	0.42	
4	2	0	0.556	0	0.444	
5	3	0	1	0	0	
6	4	0	0.667	0	0.333	
7						
8						
9						
10						

Рисунок 5.3 – Приклад структури Excel файлу



Рисунок 5.4 – Вміст завантаженого Excel файлу

Після того як користувач впевнився у тому що завантажив вірний файл може продовжити роботу натиснувши кнопку “Розрахувати матрицю переходів”.

Результати розрахунку будуть відображені в таблиці справа від завантаженого файлу ( рисунок – 5.5).

G1P8	G1Pother	Gotherp8	OtherOther
0.1	0.2	0.04	0.35
0.7	0.7	0.33	0.4
0.04	0	0.11	0.06
0.16	0.1	0.52	0.19

Рисунок 5.5 – Матриця переходів для завантажених генотипів вірусу

Після того як користувач розрахував матрицю переходів він може продовжити роботу натиснувши на кнопку “Розрахувати ефективність вакцини”. На екрані з’явиться модальне вікно (рисунок 5.6) в якому користувач має ввести ефективність вакцино проти кожної з груп гомологічного збудника (штаму збудника) а також обсяг вакцинації.

Рисунок 5.6 – Розрахунок ефективності вакцини

Після введення необхідних даних користувач має натиснути кнопку “Розрахувати”.

Результати розрахунків першого блоку програми будуть відображені графічно під таблицями (рисунок 5.7 – 5.10).

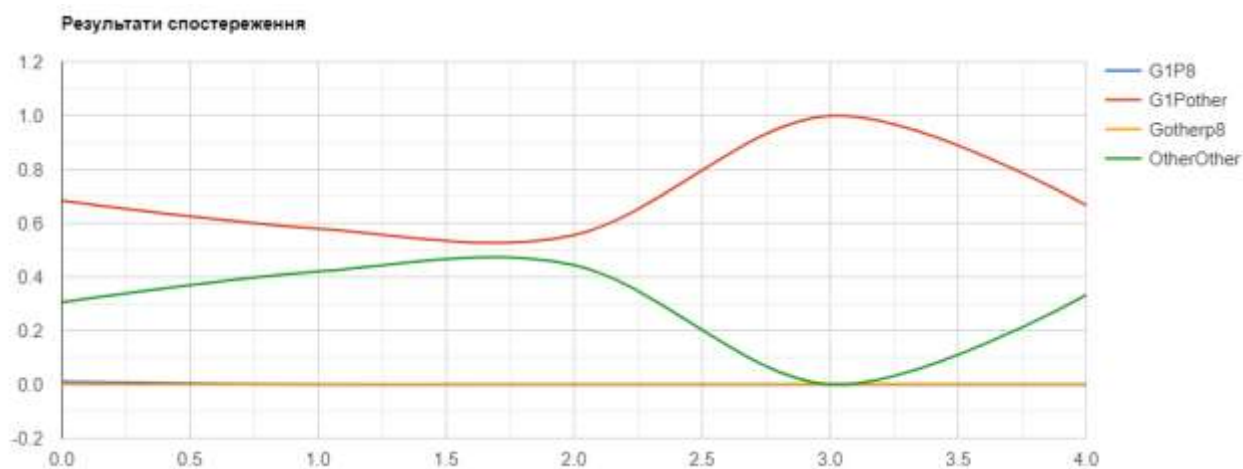


Рисунок 5.7 – Результати спостереження за вірусом

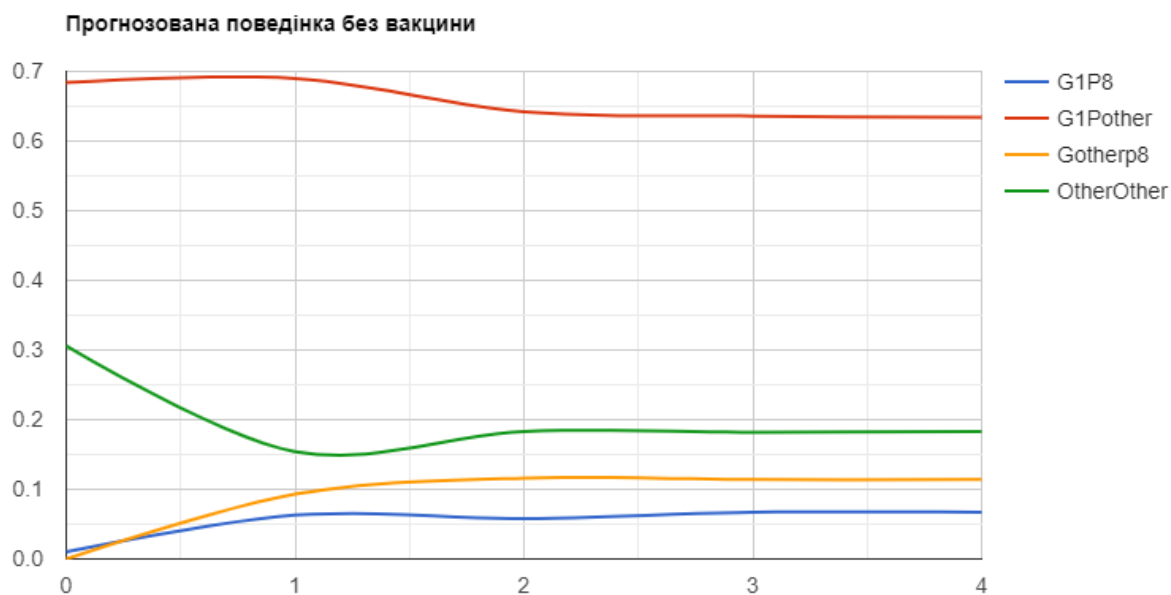


Рисунок 5.8 – Прогнозована поведінка штамів вірусу без втручання у вигляді вакцинації

На рисунку 5.8 відображається поведінка штамів вірусу яка відповідає матриці переходів зображених вище. Справа від графіка позначається колір графіка та відповідний йому штам хвороби.

Поведінка гомологічних груп вірусу під дією вакцини, параметри якої ми визначили в відповідному модальному вікні, має наступний вигляд (рисунок 5.9)

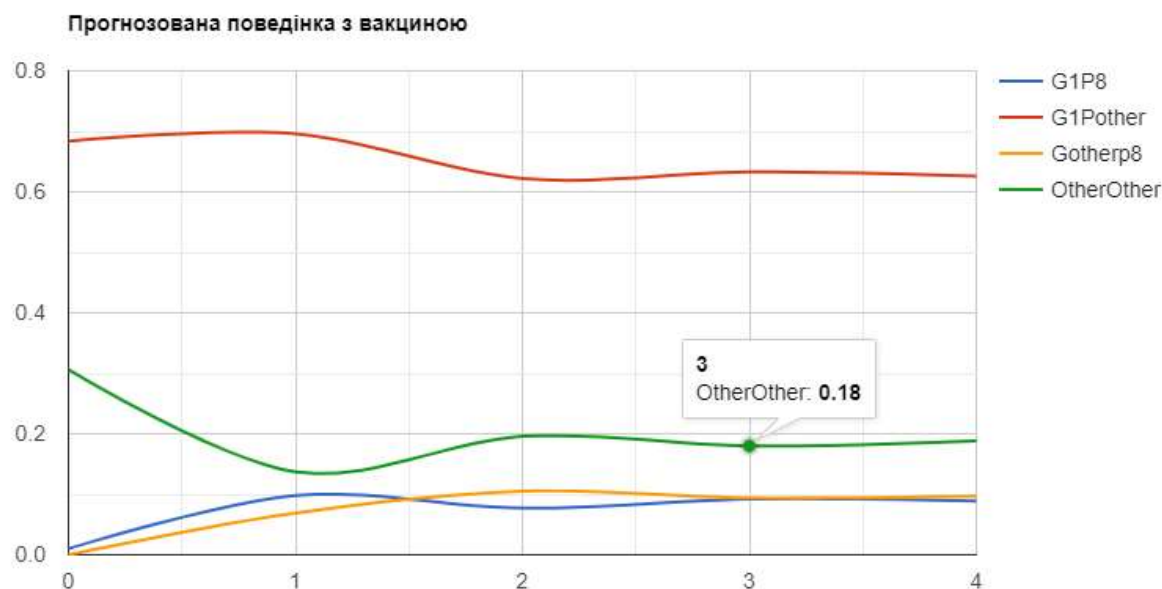


Рисунок 5.9 – Прогнозована поведінка штамів вірусу під впливом вакцинації

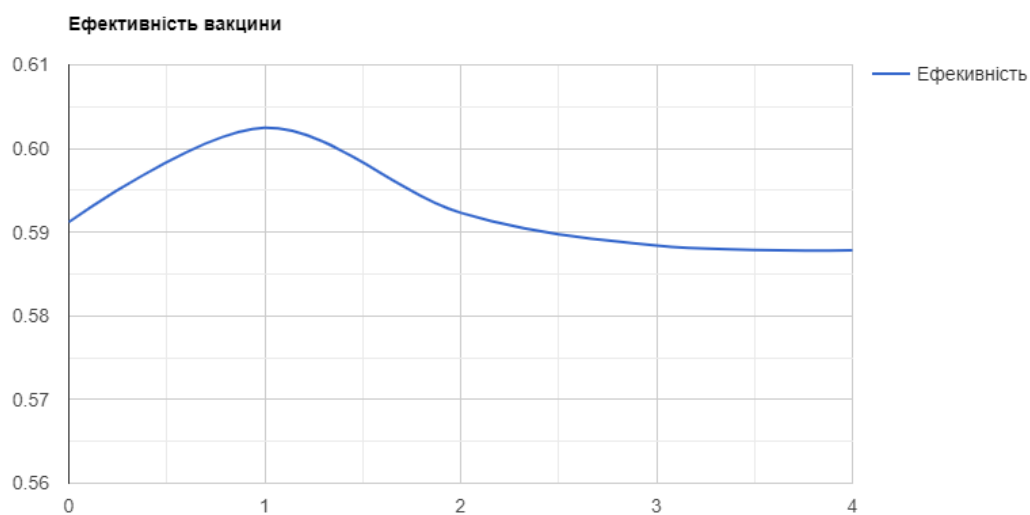


Рисунок 5.10 – Ефективність вакцини проти вірусу

Таким чином після виконання описаних вище дій у блоці аналізу динаміки збудника ми отримуємо матрицю переходів штамів інфекції, поведінку штамів інфекції

без втручання у вигляді вакцинації, поведінку штамів інфекції під дією вакцинації та ефективність вакцини проти даної інфекції.

Після цього ми переходимо до другого блоку, натискаючи на кнопку “Розрахунок параметру передачі вірусного збудника” на верхній панелі меню (рисунок 5.11).



Рисунок 5.11 – Меню розрахунку параметру передачі вірусного збудника

Для роботи з цим блоком програми нам необхідний ще один Excel файл з даними спостережень. В цьому файлі нам потрібна інформація про відносну кількість інфікованих та сприятливих до хвороби людей на певному проміжку часу. Excel файл має обов’язково містити дві колонки: “S” – відносна частина сприятливих до вірусу людей та “I” – відносна частина інфікованих людей. Дані мають бути записані в числовому форматі. Для завантаження даних користувач має натиснути кнопку “Відкрити файл зі спостережуваними даними здорових та хворих людей” після чого вибрати в провіднику потрібний файл та натиснути кнопку “Відкрити”.

Далі, користувач має розрахувати параметр передачі вірусного збудника (“Beta”) натиснувши на кнопку “Розрахувати параметр передачі збудника”. Після чого

на екран будуть виведені значення розрахованого параметру вірусного збудника (рисунок 5.12).

Вакцинопрофілактика

АНАЛІЗ ДИНАМІКИ ЗБУДНИКА    РОЗРАХУНОК ПАРАМЕТРУ ПЕРЕДАЧІ ВІРУСНОГО ЗБУДНИКА    ПОБУДОВА ЕПІДЕМІОЛОГІЧНОЇ МОДЕЛІ

Відкрити файл зі спостережуваними значеннями здорових та хворих людей    Розрахувати параметр передачі збудника

All	S	I	Beta	ConstB
1	0.999986413519863	0.0000135864801374426	1.0032981050153489	1.082030005127963
1	0.9999863688954259	0.000013631104574106897	0.8093184016161488	
1	0.9999889682466109	0.000011031753389083056	0.7840907663243255	
1	0.9999913501994547	0.000008649800545248718	1.3762918196431906	
1	0.9999880954532409	0.000011904546759168763	1.0630525566580302	
1	0.9999873449917858	0.00001265500821428626	2.3454939287202747	
1	0.9999703181306946	0.00002968186930538965	0.9915652789990191	
1	0.9999705693625632	0.000029430637436764605	1.0174395977485893	
1	0.9999700569853528	0.000029943014647193307	0.9796446637466415	
1	0.9999706673638187	0.00002933263618177298	0.8430478988017965	

Рисунок 5.12 – Розрахований параметр передачі вірусного збудника (Beta)

Також користувач зможе переглянути результати розрахунків у графічному вигляді (рисунок 5.13).

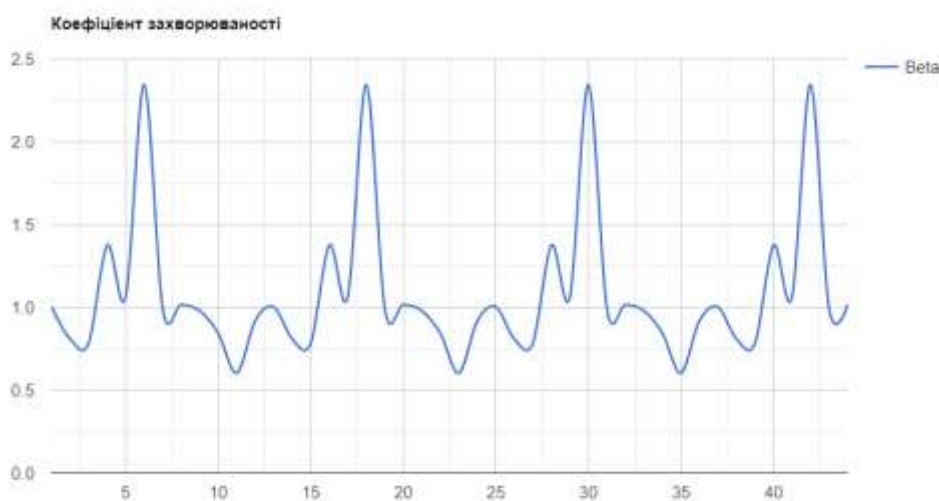


Рисунок 5.13 – Параметр передачі вірусного збудника

Після цього робота у другому блоці програми завершена.

Для того щоб перейти до третьої частини програми користувач має натиснути на кнопку “Побудова епідеміологічної моделі” (рисунок 5.14).



Рисунок 5.14 – Меню побудови епідеміологічної моделі

В цьому меню розташовані параметри епідеміологічної моделі (рисунок 5.15) які користувач має ввести в систему за допомогою чекбоксів та полів для вводу.

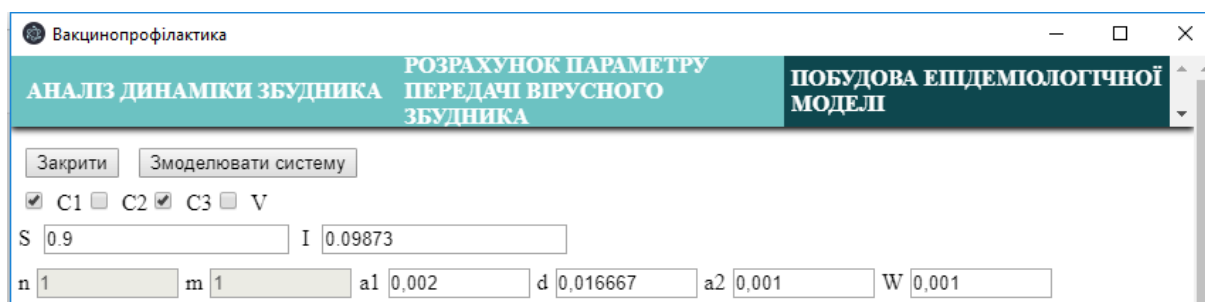


Рисунок 5.15 – Параметри епідеміологічної моделі

Перша частина – це група чекбоксів, кожен з яких впливає на початкову модель системи за якою ми будемо розраховувати епідеміологічні параметри. Перший чекбокс “C1” відповідає за те чи є система закритою чи відкритою, тобто чи буде моделі наявний притік та відтік населення на кожному кроці розрахунків. Якщо чекбокс відмічений то система є відкритою, якщо не відмічений, то закритою. Другий чекбокс “C2” відповідає за динаміку в середині системи, тобто чи

поділяється система на дві частини, в одній з яких буде задіяна вакцинація, а в іншій – ні. Якщо чекбокс “C2” відмічений, то система є розподіленою на дві підгрупи, якщо не відмічений то система моделюється з однією групою людей яка піддається вакцинації. Для того щоб визначити величину підгруп одна відносно одної, після відмічення чекбоксу ми маємо ввести два додаткові параметри “m” та “n”. Параметр “n” визначає частину сприятливого населення у підгрупі відносно всього населення, а параметр “m” частину інфікованого населення у підгрупі відносно всього населення. Ці два параметри визначають першу підгрупу населення, тобто ту яка піддається вакцинації, відповідно величина другої підгрупи буде “1-m” та “1-n”. Третій чекбокс “C3” відповідає за тип вакцинації. Якщо цей чекбокс увімкнений то вакцинація відбувається на вході людей в систему, якщо вимкнений то вакцинація відбувається в системі. Останній чекбокс “V” відповідає за те чи буде проводитися вакцинація на кожному кроці розрахунку моделі, чи лише на першому. Якщо чекбокс увімкнений то вакцинація буде проводитись на кожному кроці, якщо ні – то лише на першому.

Далі користувач має ввести значення “S” та “I”. Перше значення – це відносна частина сприятливого населення, друге – це інфіковане населення. Сума цих двох параметрів не може бути більшою за одиницю, для того щоб система працювала правильно.

Далі йдуть останні чотири параметри епідеміологічної системи. Перший параметр “a1” визначає відносну частину населення яка прибуває до системи на кожному кроці. Другий параметр “d” визначає відносну частину населення яка переходить з першої групи населення до другої на кожному кроці. Третій параметр “a2” визначає кількість людей що покидає систему на кожному кроці. Останній параметр “W” визначає яка кількість людей повернеться з несприятливого до інфекції стану (успішно вакцинована і не може бути інфікована) до сприятливого стану (може бути інфікована).

Після того як користувач коректно ввів усі необхідні параметри системи він може натиснути кнопку “Змоделювати систему” для того щоб отримати кінцевий результат.



Результат роботи представлено у вигляді таблиці з розрахованими значеннями епідеміологічної моделі (рисунок 5.16).



All Population	All illnesses	S1	I1	S2	I2	V
0.9987300000	0.0987300000	0.9000000000	0.0987300000	0.0000000000	0.0000000000	0.0000000000
1.0007274600	0.0892634243	0.9106313346	0.0892634243	0.0000000000	0.0000000000	0.0008327011
1.0018945484	0.0657384911	0.9353241889	0.0657384911	0.0000000000	0.0000000000	0.0008318684
1.0030629979	0.0481200670	0.9541118943	0.0481200670	0.0000000000	0.0000000000	0.0008310366
1.0042328101	0.0629952570	0.9404073476	0.0629952570	0.0000000000	0.0000000000	0.0008302055
1.0054039866	0.0627110678	0.9418635435	0.0627110678	0.0000000000	0.0000000000	0.0008293753
1.0065765289	0.1377925993	0.8679553837	0.1377925993	0.0000000000	0.0000000000	0.0008285459
1.0077504387	0.1178142453	0.8891084760	0.1178142453	0.0000000000	0.0000000000	0.0008277174
1.0089257176	0.1057567743	0.9023420536	0.1057567743	0.0000000000	0.0000000000	0.0008268897
1.0101023671	0.0926592496	0.9166170547	0.0926592496	0.0000000000	0.0000000000	0.0008260628

Рисунок 5.16 – Розраховані значення епідеміологічної моделі

### 4.3 Висновки до розділу 4

У даному розділі описано вимоги до технічного забезпечення користувача для забезпечення коректної роботи інформаційної системи біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій.

Описано алгоритм роботи користувача з розробленою системою.

## ВИСНОВКИ

В ході виконання роботи було розроблено інформаційну систему біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій.

Додаток було написано на мові програмування JavaScript з використанням фреймворку Electron. Інтерфейс виконано за допомогою технологій HTML та CSS.

В ході роботи було проведено аналіз предметної області, а саме вакцинопрофілактики гострих вірусних інфекцій та досліджені існуючі програмні застосунки які могли б вирішити дану проблему. Було виявлено особливості даного напрямку. Були досліджені різні типи епідеміологічних моделей, обрана та вдосконалена найпідходяща з них.

Було проведено огляд програмних засобів, використаних під час розробки системи (Visual Studio Code, мова програмування JavaScript, фреймворк Electron, технології HTML та CSS, бібліотека node-xlsx).

Для роботи з даним програмним забезпеченням користувачу достатньо буде комп'ютер, що має потужність нижче середньої. Попередньо ніякі програми встановлювати не потрібно. Під час використання додатку користувач повинен мати декілька Excel файли відповідного вмісту для забезпечення вхідних даних для розрахунків.

За допомогою цієї програми аналітики в медичній сфері зможуть слідкувати за динамікою розвитку інфекції у конкретній популяції людей з заданими параметрами, визначаючи ефективність вакцини проти вірусу та параметру передачі вірусного збудника з реальних даних спостережень. Програма може бути багаторазово використана і не накладає обмежень на тип досліджуваного вірусу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson R. Population Dynamics of Infectious Diseases: Theory and Applications / R. Anderson. — London: Chapman and Hall, 1982. — P. 5-52.
2. Tassier T. The Economics of Epidemiology / T. Tassier. — Berlin: Springer Science & Business Media, 2013. — P. 5-46.
3. Cullen, J., and West, P. 1979. The economics of health. An introduction. Oxford: MartinRobertson. Doebeli, M., C. Hauert, and T. Killingback. 2004. The evolutionary origin of cooperators and defectors. Science 306: 859–862.
4. Jansen, V.A., N. Stollenwerk, H.J. Jensen, M.E. Ramsay, W.J. Edmunds, and C.J.Rhodes. 2003. Measles out breaks in a population with declining vaccine uptake. Science 301: 804.
5. Chapman, G.B., and E.J. Coups. 2006. Emotions and preventive health behavior: Worry, regret, and influenza vaccination. Health Psychology 25: 82–90.
6. Asch, D.A., J. Baron, J.C. Hershey, H. Kunreuther, J. Meszaros, I. Ritov, and M.Spranca. 1994. Omission bias and pertussis vaccination. Medical Decision Making 14: 118–123.
7. Cullen, J., and West, P. 1979. The economics of health. An introduction. Oxford: MartinRobertson. Doebeli, M., C. Hauert, and T. Killingback. 2004. The evolutionary origin of cooperators and defectors. Science 306: 859–862.
8. Jansen, V.A., N. Stollenwerk, H.J. Jensen, M.E. Ramsay, W.J. Edmunds, and C.J.Rhodes. 2003. Measles out breaks in a population with declining vaccine uptake. Science 301: 804.
9. Bauch, C.T. 2005. Imitation dynamics predict vaccinating behavior. Proceedings of the Royal Society B 272: 1669–1675.
10. Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/>.
11. Electron [Електронний ресурс] – Режим доступу до ресурсу: <https://electronjs.org/>.

12. A. d'Onofrio, P. Manfredi, Bifurcation thresholds in an sir model with information-dependent vaccination, *Mathematical Modelling of Natural Phenomena* 2 (01) (2007) 26–43.
13. S. Bhattacharyya, C. Bauch, A game dynamic model for delayer strategies in vaccinating behaviour for pediatric infectious diseases, *Journal of Theoretical Biology* 267 (3) (2010) 276–282.
14. Fox JP, Elveback L, Scott W, et al. Herd immunity: basic concept and relevance to public health immunization practices. *Am J Epidemiol* 1971; 94 (3): 179-89.
15. Rotavirus / U.Parashar, J. Bresee, J. Gentsch [et al]// *Emerging Infectious Diseases*. – 1998. - Vol. 4. – P. 561-570. Rotavirus replication: plus-sense templates for double-stranded RNA synthesis are made in viroplasms / L.S. Silvestri, Z.F. Taraporewala, J.T. Patton // *J. Virol.* – 2004. – Vol.78. – P.7763–7774. Desselberger U. Viral factors determining rotavirus pathogenicity/ U. Desselberger // *Arch. Virol. Suppl.* – 1997. – Vol.13. – P.131–139.
16. C. T. Bauch, A. P. Galvani, D. J. Earn, Group interest versus self-interest in small pox vaccination policy, *Proceedings of the National Academy of Sciences* 100 (18) (2003) 10564–10567.
17. D. Kahneman, A. Tversky, Prospect theory: An analysis of decision under risk, *Econometrica: Journal of the Econometric Society* (1979) 263–291.

## ДОДАТОК А

Інформаційна система біоінформаційного та епідеміологічного  
аналізу вакцинопрофілактики гострих вірусних інфекцій

Специфікація

УКР.НТУУ”КПІ імені Ігоря Сікорького”\_ТЕФ\_АПЕПС\_TV5280\_19Б

Аркушів 1

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5280_19Б	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5280_19Б 12-1	index.js	Основні компоненти
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5280_19Б 13-1	Додаток В.doc	Опис програмного модуля

## ДОДАТОК Б

Інформаційна система біоінформаційного та епідеміологічного  
аналізу вакцинопрофілактики гострих вірусних інфекцій

Лістинг програми

УКР.НТУУ”КПІ імені Ігоря Сікорького”\_ТЕФ\_АПЕПС\_TV5280\_19Б

12-1

Аркушів 3

Київ 2019

```

document.getElementById('block1__rozrakhunok').addEventListener('click', () => {
    let massHTML = document.getElementsByClassName('illness-input');
    let obsiahVakcinacii = Number(document.getElementById('obsiah-input').value);
    let Efectivnosti = [];
    let massEfectivnosti = [];
    let vikMatr = objectForData.masDoli[0].vik;
    let masGOOD = [];
    let masObj = [];
    let sumEfectiv = 0;
    for(let i=0; i < massHTML.length;i++){
        let temp = Object.keys(objectForData.masDoli[0].groups[i])[0];
        let efectivnost = massHTML[i].value;
        Efectivnosti.push({
            [temp] : Number(efectivnost)
        });
    }
    console.log(Efectivnosti, obsiahVakcinacii, objectForData.bestMasPerekhody);

    masGOOD.push({
        vik : objectForData.masDoli[0].vik,
        mass : makeGroupMas(objectForData.masDoli[0].groups)
    });
    masObj.push({
        vik : objectForData.masDoli[0].vik,
        mass : makeGroupMas(objectForData.masDoli[0].groups)
    });
    for(let i=0;i < Efectivnosti.length;i++){
        let temp = Object.keys(Efectivnosti[i])[0];
        sumEfectiv += masGOOD[masGOOD.length-1].mass[i] * obsiahVakcinacii * Efectivnosti[i][temp];
    }
    massEfectivnosti.push({
        vik : objectForData.masDoli[0].vik,
        efectivnost : sumEfectiv
    });
    for(let i=1;i < objectForData.masDoli.length;){
        let vektor = [];
        for(let j=0;j<Efectivnosti.length;j++){
            let efectiv = Object.keys(Efectivnosti[j])[0];
            let temp = masObj[masObj.length-1].mass[j] * (1 - Efectivnosti[j][efectiv]) * obsiahVakcinacii + (1 -
obsiahVakcinacii) * masObj[masObj.length-1].mass[j];
            vektor.push(temp);
        }
        vektor = toOne(vektor);

        let temp = multiplyMatrix(objectForData.bestMasPerekhody,vektor);

        vikMatr++;
        masObj.push({
            vik : objectForData.masDoli[i].vik,
            mass : temp
        });
        if(vikMatr === objectForData.masDoli[i].vik){
            masGOOD.push({
                vik : objectForData.masDoli[i].vik,
                mass : temp
            });
            sumEfectiv = 0;
            for(let i=0;i < Efectivnosti.length;i++){
                let temp = Object.keys(Efectivnosti[i])[0];
                sumEfectiv += masGOOD[masGOOD.length-1].mass[i] * obsiahVakcinacii * Efectivnosti[i][temp];
            }
            massEfectivnosti.push({
                vik : objectForData.masDoli[i].vik,
                efectivnost : sumEfectiv
            });
            i++;
        }
        objectForData.massDoliVakcinirovanie = masGOOD;
    }
    objectForData.massEfectivnosti = massEfectivnosti;

```



```

deleteElem('modal_window');
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawBlock1);
function drawBlock1(){
    let masDoli = [];
    let temp = [];
    temp.push("Bik");
    for(let i=0;i<objectForData.masDoli[0].groups.length;i++){
        let temp1 = Object.keys(objectForData.masDoli[0].groups[i])[0];
        temp.push(temp1);
    }
    masDoli.push(temp);
    for(let i=0;i<objectForData.masDoli.length;i++){
        let tempMas = [];
        tempMas.push(objectForData.masDoli[i].vik);
        for(let j=0;j<objectForData.masDoli[i].groups.length;j++){
            let temp1 = Object.keys(objectForData.masDoli[i].groups[j])[0];
            tempMas.push(objectForData.masDoli[i].groups[j][temp1]);
        }
        masDoli.push(tempMas);
    }

    let masDoliCalibrated = [];
    let tempCalibr = [];
    tempCalibr.push("Bik");
    for(let i=0;i<objectForData.masDoli[0].groups.length;i++){
        let temp1 = Object.keys(objectForData.masDoli[0].groups[i])[0];
        tempCalibr.push(temp1);
    }
    masDoliCalibrated.push(tempCalibr);
    for(let i=0;i<objectForData.masDoliCalibrated.length;i++){
        let tempMas = [];
        tempMas.push(objectForData.masDoliCalibrated[i].vik);

        for(let j=0;j<objectForData.masDoliCalibrated[i].mass.length;j++){
            let temp1 = Object.keys(objectForData.masDoliCalibrated[i].mass[j])[0];
            tempMas.push(objectForData.masDoliCalibrated[i].mass[j][temp1]);
        }
        masDoliCalibrated.push(tempMas);
    }

    let masDoliVac= [];
    let tempVac = [];
    tempVac.push("Bik");
    for(let i=0;i<objectForData.masDoli[0].groups.length;i++){
        let temp1 = Object.keys(objectForData.masDoli[0].groups[i])[0];
        tempVac.push(temp1);
    }
    masDoliVac.push(tempCalibr);
    for(let i=0;i<objectForData.massDoliVakcinirovanie.length;i++){
        let tempMas = [];
        tempMas.push(objectForData.massDoliVakcinirovanie[i].vik);

        for(let j=0;j<objectForData.massDoliVakcinirovanie[i].mass.length;j++){
            let temp1 = Object.keys(objectForData.massDoliVakcinirovanie[i].mass[j])[0];
            tempMas.push(objectForData.massDoliVakcinirovanie[i].mass[j][temp1]);
        }
        masDoliVac.push(tempMas);
    }

    let masEffect= [];
    let tempEffect = [];
    tempEffect.push("Bik");
    tempEffect.push("Ефективність");
    masEffect.push(tempEffect);
    for(let i=0;i<objectForData.massEffectivnosti.length;i++){
        let tempMas = [];
        tempMas.push(objectForData.massEffectivnosti[i].vik);
        tempMas.push(objectForData.massEffectivnosti[i].effectivnost)
        masEffect.push(tempMas);
    }

```

```

    }

    const data = google.visualization.arrayToDataTable(masDoli);
    const options = {
        title : "Результати спостереження",
        curveType: 'function',
        legend: {position:'right'},
        width: 1200,
        height: 500
    }
    const pie_1_chart = new google.visualization.LineChart(document.getElementById('chart1'));
    pie_1_chart.draw(data, options);

    const data1 = google.visualization.arrayToDataTable(masDoliCalibrated);
    const options1 = {
        title : "Прогнозована поведінка без вакцини",
        curveType: 'function',
        legend: {position:'right'},
        width: 900,
        height: 500
    }
    const pie_2_chart = new google.visualization.LineChart(document.getElementById('chart2'));
    pie_2_chart.draw(data1, options1);
    const data2 = google.visualization.arrayToDataTable(masDoliVac);
    const options2 = {
        title : "Прогнозована поведінка з вакциною",
        curveType: 'function',
        legend: {position:'right'},
        width: 900,
        height: 500
    }
    const pie_3_chart = new google.visualization.LineChart(document.getElementById('chart3'));
    pie_3_chart.draw(data2, options2);

    const data3 = google.visualization.arrayToDataTable(masEffect);
    const options3 = {
        title : "Ефективність вакцини",
        curveType: 'function',
        legend: {position:'right'},
        width: 900,
        height: 500
    }
    const pie_4_chart = new google.visualization.LineChart(document.getElementById('chart4'));
    pie_4_chart.draw(data3, options3);
    }
    });
});

```

## ДОДАТОК В

Інформаційна система біоінформаційного та епідеміологічного  
аналізу вакцинопрофілактики гострих вірусних інфекцій

Опис програми

УКР.НТУУ”КПІ імені Ігоря Сікорького”\_ТЕФ\_АПЕПС\_TV5280\_19Б

13-1

Аркушів 9

Київ 2019

## АНОТАЦІЯ

Даний додаток містить опис двох функцій програми які слугують засобом для зчитування, обробки та виведення інформації щодо поведінки штамів вірусної інфекції та розрахунку ефективності вакцини проти даної інфекції. Модуль надає можливість розрахунку очікуваної поведінки штамів вірусу без втручання у вигляді вакцинації та з вакцинацією. Розраховує ефективність вакцини по введеним користувачем даним про ефективність вакцини проти кожного штаму вірусу. Відображає описані вище розрахунки у вигляді графіків. Модуль написано на мові програмування JavaScript, з використанням бібліотеки node-xlsx та бібліотеки Google Charts.

## ЗМІСТ

1. Загальні відомості.....	3
2. Функціональне призначення.....	4
3. Опис логічної структури .....	5
4. Технічні засоби, що використовуються .....	6
5. Виклик і завантаження .....	7
6. Вхідні і вихідні дані.....	8

## ЗАГАЛЬНІ ВІДОМОСТІ

У додатку розглянуто модуль розрахунку ефективності вакцини проти хвороби з кодом УКР.НТУУ”КПІ імені Ігоря Сікорького”\_ТЕФ\_АПЕПС\_TB5280\_19Б 12-1, що міститься у файлі index.js. Модуль реалізовано мовою JavaScript, бібліотек node-xlsx та Google Charts. Модуль призначений для розрахунку ефективності вакцини. Користувач отримує матрицю переходів штамів віусу та дані про ефективність вакцини в числовому та графічному вигляді.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблені компоненти виконують завдання зчитування даних з Excel файлу, відображення цих даних. На основі зчитаних даних розраховує матрицю переходів штамів вірусу. Визначає очікувану поведінку штамів вірусу відповідно до матриці переходів.

Далі на основі введених користувачем даних про ефективність вакцини проти штамів вірусу розраховує визначає їх поведінку під дією вакцини та розраховує ефективність вакцини проти даного вірусу. Графічно зображує результат.

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

При запуску програми користувач бачить головне вікно яке має переходи до трьох блоків програми, в яких може проаналізувати динаміку збудника, розрахувати параметр передачі вірусного збудника та побудувати епідеміологічну модель.



## **ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ**

Для створення програмного застосунку було використано середовище розробки Visual Studio Code, мова програмування JavaScript та фреймворк Electron.

Також використовуються технології HTML, CSS, бібліотеки node-xlsx та Google Charts.

## **ВИКЛИК І ЗАВАНТАЖЕННЯ**

Розроблена система не потребує інсталяції, треба запустити виконуваний файл під назвою “Vаксунoprofilactyca”.

Після завантаження перед користувачем відкриється головне меню програми звідки він може розпочати роботу.

## **ВХІДНІ І ВИХІДНІ ДАНІ**

Вхідними даними є інформація яка зчитується з Excel файлів та дані які користувач вводить до “input” полів.

Вихідними даними програми є числова інформація у вигляді таблиць та графіки функцій.

## ДОДАТОК Г

Інформаційна система біоінформаційного та епідеміологічного  
аналізу вакцинопрофілактики гострих вірусних інфекцій

Апробації

УКР.НТУУ”КПІ імені Ігоря Сікорького”\_ТЕФ\_АПЕПС\_TV5280\_19Б

Аркушів 3

Київ 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ГОРЯ СІКОРСЬКОГО»

# СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної  
науково-практичної конференції  
молодих вчених та студентів  
м. Київ, 23-26 квітня 2019 року,

ТОМ 1



Київ- 2019

<b>Природні ресурси регіону. Система обліку лісових ресурсів.</b>	164
<i>ЛІНЬОВ Д.О., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Інтернет-система редагування та аналізу генетичних послідовностей.</b>	165
<i>ЖИРОВ М.І., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Google maps API. Обробка даних отриманих з пристрою та їх відображення на карті веб-інтерфейсу.</b>	166
<i>ЖИРНОВ А.Ю., студент гр. ТВ-51</i>	
<i>Керівник - доц., к.т.н. Смаковський Д.С.</i>	
<b>Природні ресурси регіону. Система обліку водних ресурсів.</b>	167
<i>ДАВИДЧУК О.С., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Інформаційна система біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій.</b>	168
<i>ГОРБАТЮК О.І., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення та прогнозу статусу гравця/команди гравців.</b>	169
<i>ГОНЧАР О.В., студент гр. ТР-52</i>	
<i>Керівник - доц., к.т.н. Коваль О.В.</i>	
<b>Природні ресурси регіону. Система обліку лікувальних ресурсів.</b>	170
<i>ГАРКУША О.В., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Система управління процесом утворення водню з відновлюваної целюлозовмісної сировини у біореакторі.</b>	171
<i>БОНДАРЕНКО А.В., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Автоматизована система для моделювання та аналізу адміністративних функцій організацій.</b>	172
<i>БІТИК М.О., студент гр. ТМ-51</i>	
<i>Керівник - доц., к.т.н. Кузьмін В.О.</i>	
<b>Інформаційна система ефективного менеджменту роботи закладів післядипломної освіти.</b>	173
<i>БАСАЛІК Д.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Інформаційна система аналізу профілактики хронічних вірусних інфекцій.</b>	174
<i>БАСАЛІК Г.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Розробка сайту для продажу продуктів здорового харчування.</b>	175
<i>СЕМІОН В.І., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Організація по впровадженню та обслуговуванню автоматизованого обладнання теплиць.</b>	176
<i>ЛОКОТАРЬОВ С.О., студент гр. ТМ-62;</i>	
<i>ШАПОВАЛ В.О., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Розробка прикладного програмного забезпечення для автоматизації роботи кінологічного клубу.</b>	177
<i>ЛЕБЕДИК Т.О., студент гр. ТМ-62;</i>	
<i>САПОН О.М., студент гр. ТМ-62</i>	

УДК 621.43.056:632.15

Студент 4 курсу, гр. ТР-52 Горбатюк О.І.  
Ст. викл. Бандурка О.І.**ІНФОРМАЦІЙНА СИСТЕМА БІОІНФОРМАЦІЙНОГО ТА  
ЕПІДЕМІОЛОГІЧНОГО АНАЛІЗУ ВАКЦИНОПРОФІЛАКТИКИ ГОСТРИХ  
ВІРУСНИХ ІНФЕКЦІЙ**

Одним з найважливіших завдань вірусології та епідеміології є прогнозування поширення, розвитку та наслідків вірусних інфекцій. Щороку в Україні реєструють близько 6 млн інфекційних захворювань, 98% з яких припадає на грип та ГРВІ. Програма вакцинації конкретною вакциною в певній віковій групі чи групі ризику скорочує число випадків захворювання[1].

Крім того, існує популяційний імунітет, тобто певна особа, яка не вакцинована, може не бути інфікованою. Взагалі, існує конфлікт між індивідуальними та соціальними вигодами від вакцинації; словом, конфлікт між індивідуальним раціональним вибором – намагатися уникнути вакцинації, або погодитися на неї, досягнувши справедливого оптимального паритету. Вакцинацію в такому випадку можна розглядати як гру в складній соціальній мережі[2].

Вакцинація, зазвичай є добровільною, тому прийняття рішень на індивідуальному рівні може бути результатом компромісу між захистом та усвідомленням ризиків й витрат на вакцинацію[3].

Програмний застосунок даватиме можливість моделювання епідеміологічної моделі для певної групи населення або популяції. Епідеміологічні моделі дозволяють досліджувати процес розповсюдження інфекційних захворювань, складати прогноз на майбутнє, визначати ефективність профілактичних заходів. Завдяки чому стає можливим своєчасне попередження епідемій інфекційних захворювань.

Основний принцип побудови епідеміологічної моделі для конкретного інфекційного захворювання: все населення ділиться на декілька категорій, модель є динамічною системою, яка змінюється в часі, тому при проходженні певного часового періоду частина населення переміщується з однієї категорії в іншу. Перехід осіб із однієї категорії в іншу характеризується за допомогою параметрів, наприклад: сила інфекції, швидкість одужання, народжуваність, смертність, міграція, тощо. Отримана епідеміологічна модель є інтегрованою, що дозволяє використати її для прогнозування захворювань на визначений період[4].

Перелік посилань:

1. Чому потрібно зробити щеплення проти грипу і як пройти вакцинацію [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <http://moz.gov.ua/article/health/chomu-potribno-zrobiti-sheplennja-proti-gripu-i-jak-projti-vakcinaciju>
2. Chapman, G.B., and E.J. Coups. 2006. Emotions and preventive health behavior: Worry, regret, and influenza vaccination. *Health Psychology* 25: 82–90
3. Asch, D.A., J. Baron, J.C. Hershey, H. Kunreuther, J. Meszaros, I. Ritov, and M. Spranca. 1994. Omission bias and pertussis vaccination. *Medical Decision Making* 14: 118–123
4. Hershey John C, David A. Asch, Thi Thumasathit, Jaqueline Meszaros and Victor V. Walters. *Organizational Behavior and Human Decisions Processes* 59, 177-187, 1994

